УДК 681.3

СИНТЕЗ МУЛЬТИСТРУКТУРНЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

А.М. Катина, С.В. Шидловский

Томский государственный университет систем управления и радиоэлектроники
E-mail: alenakatina@yandex.ru, stas@iit.tusur.ru

Рассматриваются основные принципы и подходы к построению мультиструктурных систем на базе лингвистического процессора. Привлечен математический аппарат теории нечетких множеств для построения имитационной модели лингвистической системы.

В настоящее время значительное повышение эффективности управления в различных сферах человеческой деятельности достигнуто за счет использования интеллектуальных систем, и ощутимые результаты получены с помощью имитационного моделирования. Особую значимость приобрели системы имитационного моделирования, основанные на знаниях. Последние являются порождением двух технологий: экспертных систем и имитационного моделирования, что позволило встраивать в разрабатываемое программное обеспечение большую часть необходимого знания и экспертизы. Использование таких систем предоставляет разработчику выполнение имитационных исследований корректно и без специального обучения в различных областях знаний.

Применение современных подходов и новейших средств вычислительной техники привело к зарождению «экспериментирующей теоретической лингвистики»: той, которая пытается построить действующие модели языка и речи не для ограниченного языкового материала, а для всей языковой способности человека в целом, на каком бы языке он ни говорил [1].

Существующие модели понимания в вычислительной лингвистике ограничены, главным образом, из-за отставания от современного уровня теоретического языкознания. Попытки формализовать интеллектуальную деятельность человека привели к постановке фундаментальной лингвистической задачи, состоящей в моделировании его языкового поведения, т. е. построения функциональной кибернетической модели естественного языка (ЕЯ). Если обозначить множество текстов через (Т), а множество выражаемых ими смыслов через (С), то модель ЕЯ можно определить как транслятор, устанавливающий соответствие между этими двумя множествами: (Т)$\to$(С) [2].

Формальные модели языка, разрабатывающиеся в последнее время, все чаще рассматриваются как компоненты различных прикладных систем. Один из классов таких систем образуют информационно-поисковые системы (ИПС), ориентированные на естественно-языковое общение с пользователем. Подобные ИПС могут использоваться в качестве консультанта – например, в области законодательства, медицины или любой другой предметной области, для которой характерно наличие большого количества информации, представленной документами на естественном языке. Поэтому разработка систем, способных понимать ЕЯ считается «основной целью исследований в области искусственного интеллекта».

Вычислительную среду, реализующую формальную лингвистическую модель и способную работать с ЕЯ во всем его объеме, будем называть лингвистическим процессором (ЛП). В современной информатике лингвистическими процессорами называются и другие средства переработки текстовой информации на ЕЯ, в том числе и не рассчитанные на работу с ЕЯ в полном объеме.

Определим две основные функции ЛП:

- извлечение смысла из заданного текста (моделирование понимания, анализ);
- выражение заданного смысла текстом на ЕЯ (моделирование производства текстов, синтез).

Разрабатываемый лингвистический процессор может быть выполнен в виде библиотеки, доступной различным приложениям и не взаимодействующей напрямую с пользователем. Однако выбранный способ построения синтаксического анализатора требует, чтобы этот блок был выполнен в виде отдельной библиотеки. Таким образом, реальная структура разработанного продукта имеет вид, представленный на рис. 1.

Заметим, что связь между модулями DEMO.EXE и LP.DLL реализуется статическим импортом, а между LP.DLL и SYNT.DLL – динамическим, что позволяет модифицировать SYNT.DLL без выхода из программы DEMO.EXE.

В рассматриваемом примере используется простейший лексический анализатор, он выполняет вспомогательные функции, не имеет возможности настройки или взаимодействия с пользователем и поэтому реализован как часть блока морфологического анализа.

Блок лексического анализа принимает исходный текст непосредственно от элементов пользовательского интерфейса – а именно, от текстового редактора. Анализируемое предложение попадает на вход лексического анализатора в виде массива ASCII-символов, содержащего прописные и строчные буквы русского алфавита, цифры, знаки пунктуации. Указатели на все сформированные таким образом строки анализатор заносит в выходной динамический массив, который является результатом его работы.

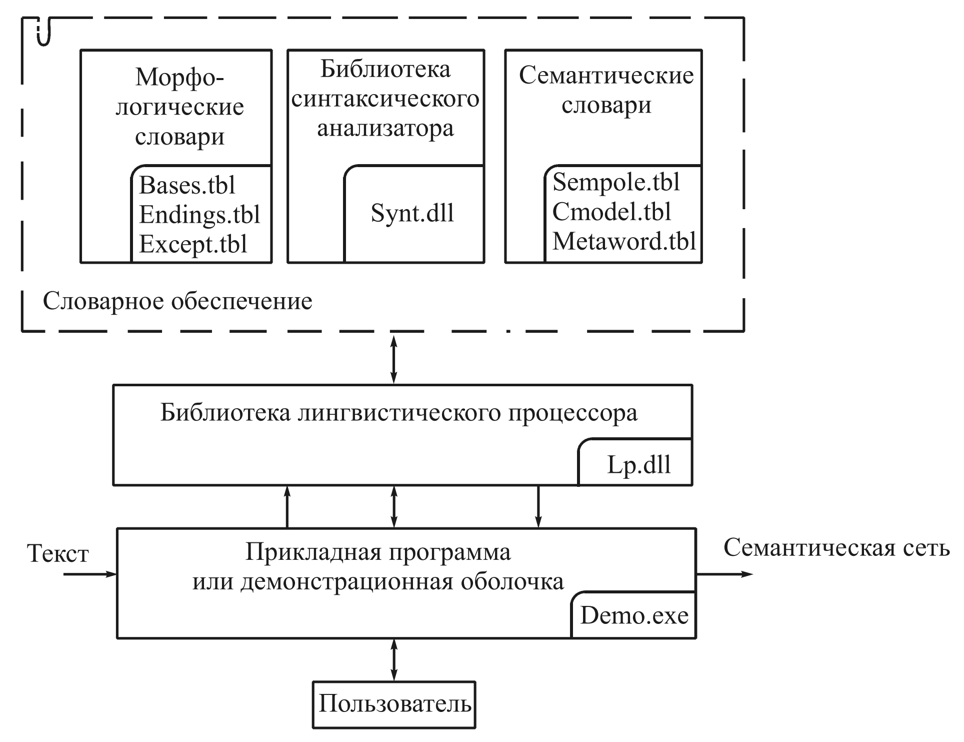


Рис. 1. *Структурная схема ЛП для поисковой системы*

Среди методов морфологического анализа, использующихся в лингвистических процессорах, можно выделить методы с декларативной и с процедурной ориентацией. Для методов декларативной ориентации характерно наличие полного словаря всех возможных словоформ для каждого слова. При этом каждая словоформа снабжается полной и однозначной морфологической информацией, куда входят как постоянные, так и переменные морфологические параметры. Задача морфологического анализа в этом случае сводится к поиску нужной словоформы в словаре и копированию морфологической информации, соответствующей найденной словоформе, в программу. В процедурных методах каждое слово разделяется на основу и аффикс (окончание и, возможно, суффикс), и словарь содержит только основы слов вместе со ссылками на соответствующие строки в таблице возможных аффиксов. Блок морфологического анализа реализован в файле MORPHO.PAS, входящем в состав исходных текстов модуля LP.DLL. В этом файле присутствуют не только функции морфологического анализа, но и средства поддержки и редактирования словарных файлов, необходимых анализатору. Файлы ENDINGS.TBL и EXEPTION.TBL также загружаются в память во время инициализации блока морфологического анализа. В памяти они хранятся в виде двух массивов текстовых строк, считанных из соответствующих файлов.

Блок синтаксического анализа реализован в файлах SYNTAX.PAS и RULES.PAS. Первый из них реализует поддержку общих механизмов синтаксического анализатора, а второй – описания конкретных правил, используемых при анализе. Таким образом, файл RULES.PAS можно рассматривать как своего рода «словарь» правил синтагматики.

Анализатор позволяет использовать две основных стратегии, отличающиеся последовательностью применения правил – а значит, и последовательностью обхода дерева вариантов разбора.

При анализе «по приоритетам» на очередной итерации выбирается одно рабочее правило и делается попытка его применения поочередно в каждой позиции анализируемого предложения. Как только применение правила окажется успешным, перебор правил начинается сначала. Последовательность использования правил анализа задается жестко с помощью системы приоритетов: чем выше приоритет данного правила, тем раньше оно будет испробовано. Отдельные правила можно временно «выключать» из анализа, устанавливая их приоритет равным нулю.

Другая стратегия анализа – «по порядку слов» – опирается на последовательность слов в анализируемом предложении. На очередной итерации устанавливается фиксированная рабочая позиция в предложении, и в этой позиции поочередно испытываются все правила анализа. Как только одно из правил сработает, обход предложения начинается сначала. (Последовательность применения правил в рабочей позиции определяется той же системой приоритетов.) Начинается работа синтаксического анализатора с инициализации списка правил (т. е. его сортировки по приоритетам) и очистки протокола. Протокол – вспомогательная структура данных типа стек, которую анализатор использует для регистрации формируемых им текстовых единиц. Каж-

дая сформированная текстовая единица (ТЕ) потенциально является узлом дерева альтернатив синтаксической структуры. Регистрация этих узлов позволяет гарантировать, что никакая ветвь дерева альтернатив не будет пропущена или пройдена дважды.

Поскольку синтаксический анализ больших предложений может занимать достаточно продолжительное время, механизм анализатора предусматривает возможность принудительного ограничения времени анализа. Если к моменту истечения этого времени сформировать синтаксическую структуру предложения не удалось, анализатор выдает на выход «лучший» из всех частичных вариантов, которые ему удалось найти. В качестве лучшего берется вариант, содержащий наименьшее число текстовых единиц верхнего уровня.

Наиболее простым и универсальным средством представления знаний в системах искусственного интеллекта является семантическая сеть. В общем случае она представляет собой ориентированный граф, вершины которого обозначают сущности (объекты), а ребра – отношения (связи) между ними. Имена вершин и ребер обычно совпадают с именами соответствующих сущностей и отношений, используемыми в естественном языке. Ребро и две связываемые им вершины представляют минимальную смысловую информацию – факт наличия связи определенного типа между соответствующими объектами.

По роду хранимой информации выделяют два типа семантических сетей: А-сети (концептуальные, интенсиональные) и К-сети (фактуальные, экстенсиональные). Первые содержат множество объектов и отношений, *допустимых* в данной предметной области; вторые – множество объектов и отношений, *присутствующих* в описании конкретной ситуации.

Для выделения в семантической сети некоторых законченных фрагментов используется понятие *семантического пространства*. Это понятие аналогично скобкам в математике. Если представить семантическую сеть в виде кортежа

$$S=<V, U>,$$

где V – множество вершин, U – множество ребер, то семантическое пространство можно обозначить как

$$P=<W, U'>,$$

где $W=\{V',\{P_i\}\}$, $U' H U$, $V' H V$, $\{P_i\}$ – множество всех семантических пространств, вложенных в пространство P.

Задачей семантического анализатора ЛП является преобразование синтаксического дерева зависимостей в соответствующий фрагмент К-сети. При этом может выполняться проверка допустимости каждого семантического отношения по опорной А-сети.

Из множества слов русского языка по семантическим признакам можно выделить следующие категории рис. 2.

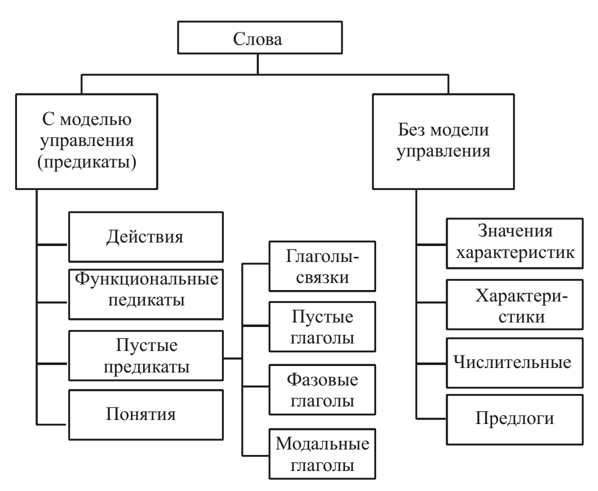


Рис. 2. *Классификация слов русского языка по семантическим категориям*

Для лингвистического процессора, предназначенного для работы в составе ИПС, наиболее важной является обработка предикатов действия и их возможных актантов – понятий, характеристик и их значений, а также числительных. Обработка функциональных и пустых предикатов может быть сведена к модификации фрагментов семантической сети, построенных для зависящих от них предикатов действия. Словарная поддержка семантического анализа включает в себя: словарь моделей управления (файл CMODEL.TBL), словарь метапонятий (файл METAWORD.TBL) и словарь семантических ролей (файл SEMROLE.TBL).

Поиск в базе данных выполняется на основе совпадения (или некоторого соответствия) семантических структур пользовательского запроса и хранимого документа. Организованный таким способом поиск (будем называть его *семантическим*) обеспечивает гораздо более высокую точность и полноту, чем традиционный поиск по ключевым словам. Однако реализация семантического поиска, очевидно, является существенно более трудоемкой, чем реализация поиска по ключевым словам. Наиболее сложным для разработки элементом здесь является именно лингвистический процессор.

Обычно говорят о понимании текстов в слабом или сильном смысле. Понимание в слабом смысле имеет место только тогда, когда обрабатываемый текст может быть перефразирован средствами того же самого или другого языка. Это обстоятельство нашло отражение в системах, использующих нечеткие правила. Неясность, неопределенность или неточность, заключенные в смысловых значениях или выводах, присущи естественным языкам. В общем случае различают три типа неточностей:

– неопределенность, когда некоторое понятие применимо к множеству разнообразных ситуаций;
– неоднозначность, когда оно описывает несколько различимых понятий;
– неясность, когда нет точно определенных границ понятий.

Нечеткое множество служит представлением всех трех типов нечеткостей.

Для моделирования понимания в обоих смыслах в любых системах, претендующих на глубокую переработку естественно-языковых текстов, необходимо иметь особый уровень представления высказываний, который можно назвать семантическим. Он задается формальным семантическим языком. Для того чтобы соотнести с вычислительной системой такие виды умственной деятельности человека, как размышление и понимание, нужно решить следующие задачи:

– уметь хорошо представлять с помощью абстрактных моделей, отражающих семантику объекта, знания и способности человека к решению задач;
– описать семантику с помощью хорошо определенного синтаксиса.

В противоположность абстракции семантика должна быть реалистичной и прагматичной, чтобы с ее помощью можно было справиться с конкретной ситуацией в реальном мире.

Одним из вариантов создания имитационной лингвистической модели системы, входящей в состав ЛП, может быть следующая реализация.

Пусть поведение рассматриваемой динамической системы описывается дифференциальным уравнением

$$\frac{dy}{dt} = f(t, y, \rho), \qquad (1)$$

где ρ – вектор параметров правой части уравнения. В общем случае этот вектор не известен и может изменяться по неопределенному закону. На практике часто известно множество R, которому принадлежат возможные значения $\rho \in R$. В этом случае целесообразно заменить ур. (1) на дифференциальное включение

$$\frac{dy}{dt} = F(t, y, R). \qquad (2)$$

Если различные точки множества Ω как возможные реализации R неравноправны, то множество R можно считать нечетким [3]. По принципу обобщения Заде функция $f(t,x)$ продолжается на семейство нечетких множеств $F(\cdot)$, т. е. в правой части (2) получаем нечеткое множество. Таким образом, можно ввести понятие решения нечеткого дифференциального включения типа (2) через понятие множества уровня.

Мультиструктура ЛП представляет собой многоуровневый преобразователь. В нем различают три уровня пофразного представления текста – морфологический, синтаксический и семантический. Каждый из уровней обслуживается соответствующими компонентами модели – массивом правил и определенным словарем или словарями. На каждом из уровней предложение имеет формальный образ, именуемый в дальнейшем его структурой- морфологической (МорфС), синтаксической (СинтС) и семантической (СемС). В целом работу модели при анализе можно представить структурной схемой, представленной на рис. 3. Синтез представляет собой обратный переход от СемС предложения к его записи в обычном орфографическом виде.

Под морфологической структурой понимается последовательность входящих в анализируемое предложение слов с указанием части речи и морфологических характеристик (падежа, числа, рода, одушевленности и т. п.).

Под синтаксической структурой понимается дерево зависимостей, в узлах которого стоят слова данного естественного языка с указанием части речи и грамматических характеристик, а дуги соответствуют специфичным для данного естественного языка отношениям синтаксического подчинения.

Под семантической структурой понимается дерево зависимостей, в узлах которого стоят либо предметные имена, либо слова универсального семантического языка.

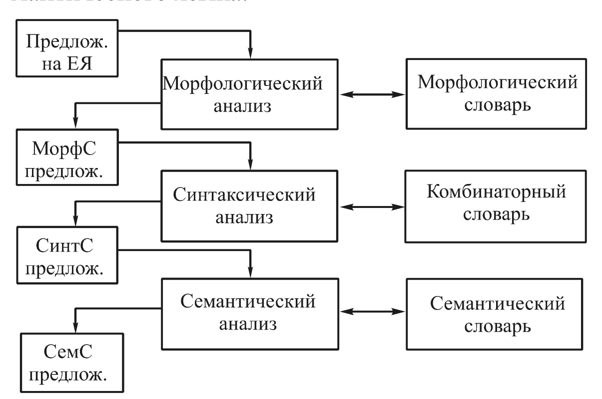


Рис. 3. *Структурная схема работы модели*

Таким образом, чтобы построить ЛП указанного типа, необходимо разработать:

– формальные языки для записи (образов) предложений на морфологическом, синтаксическом и семантическом уровнях представления;
– формальное понятие структуры предложения для каждого из этих уровней;
– массивы правил для преобразования структур смежных уровней друг в друга;
– морфологический, комбинаторный и семантический словари, включив в них всю информацию о каждой лексеме, необходимую для осуществления соответствующего преобразования.

Для реализации многоязычного ЛП, такую работу следует выполнить для каждого из участвующих в нем ЕЯ, а чтобы получить полифункциональный ЛП, необходимо постоянно пополнять его средствами решения каждой очередной задачи, если они специфичны для нее. Так, для автоматизации перевода с одного ЕЯ на другой ЛП должен быть дополнен соответствующим массивом правил перевода.

СПИСОК ЛИТЕРАТУРЫ

1. Демьянков В.З. Интерпретация, понимание и лингвистические аспекты их моделирования на ЭВМ. – М.: Изд-во Моск. ун-та, 1989. – 172 с.
2. Тей А. и др. Логический подход к искусственному интеллекту: от классической логики к логическому программированию: Пер. с франц. – М.: Мир, 1990. – 432 с.
3. Петров Б.Н., Уланов Г.М., Гольденблат И.И., Ульянов С.В. Теория моделей в процессе управления. – М.: Наука, 1978. – 224 с.

УДК 621.376.9